ELSEVIER
Available online at www.sciencedirect.com
SCIENCE DIRECT
Journal of Complexity 21 (2005) 757–771
Journal of COMPLEXITY
www.elsevier.com/locate/jco

Lower bounds for wrap-around L_2-discrepancy and constructions of symmetrical uniform designs

Kai-Tai Fang[a,*], Yu Tang[a,b], Jianxing Yin[b]

[a] *Department of Mathematics, Hong Kong Baptist University, Hong Kong, PR China*
[b] *Department of Mathematics, Suzhou University, Suzhou 215006, PR China*

Received 17 May 2004; accepted 18 January 2005

Abstract

The wrap-around L_2-discrepancy has been used in quasi-Monte Carlo methods, especially in experimental designs. In this paper, explicit lower bounds of the wrap-around L_2-discrepancy of U-type designs are obtained. Sufficient conditions for U-type designs to achieve their lower bounds are given. Taking advantage of these conditions, we consider the perfect resolvable balanced incomplete block designs, and use them to construct uniform designs under the wrap-around L_2-discrepancy directly. We also propose an efficient balance-pursuit heuristic, by which we find many new uniform designs, especially with high levels. It is seen that the new algorithm is more powerful than existing threshold accepting ones in the literature.

Keywords: Uniform designs; Lower bound; Wrap-around L_2-discrepancy; Perfect RBIBD

1. Introduction

Recently, Fang et al. [6] gave lower bounds for the wrap-around L_2-discrepancy (WD_2, for simplicity) of U-type designs with two or three levels and made a significant improvement to the threshold accepting algorithm. In this paper, we extend their results to any symmetrical U-type designs, define the perfect resolvable balanced incomplete block designs, propose a more efficient algorithm, namely balance-pursuit heuristic, for searching uniform designs and obtain new uniform designs with more levels.

* Corresponding author.
E-mail address: ktfang@hkbu.edu.hk (Kai-Tai Fang).

doi:10.1016/j.jco.2005.01.003

For more than two decades since the uniform design was proposed by Fang and Wang [2,16], it has been widely applied in manufacturing, system engineering, pharmaceutics and natural sciences. Uniform design is a type of "space filling" designs, which spread experimental points uniformly on the experimental domain [1]. The construction methods of uniform designs by most authors in the literature have been restricted on balanced lattice designs, or called *U-type designs*. In this paper, we only consider symmetrical U-type designs. A symmetrical U-type design $U(n; q^m)$ corresponds to an $n \times m$ matrix $X = (x_1, \ldots, x_m)$ such that each column x_i takes values from a set of q integers, say $\{1, 2, \ldots, q\}$, equally often. Denote by $\mathcal{U}(n; q^m)$ the set of all $U(n; q^m)$ designs. By mapping $f: l \to (2l-1)/(2q), l = 1, \ldots, q$, the n runs are transformed into n points in $C^m = [0, 1]^m$. The transformed design is denoted by $\tilde{U}(n; q^m)$ and the set of all such designs is denoted by $\tilde{\mathcal{U}}(n; q^m)$. The one-to-one correspondence between $\mathcal{U}(n; q^m)$ and $\tilde{\mathcal{U}}(n; q^m)$ will be used often throughout this paper.

As a measure of uniformity, the L_p-discrepancy has been widely used in quasi-Monte Carlo methods (see [9,15]). However, in [10,11], Hickernell pointed out some weakness of the L_p-discrepancy and further proposed several modifications, among which the wrap-around L_2-discrepancy (WD_2) is an attractive and interesting one. Let $\mathcal{P} = \{(x_{k1}, \ldots, x_{km}), k = 1, \ldots, n\}$ be a set of n points in C^m. An analytical expression of $WD_2(\mathcal{P})$ can be derived.

$$(WD_2(\mathcal{P}))^2 = -\left(\frac{4}{3}\right)^m + \frac{1}{n}\left(\frac{3}{2}\right)^m + \frac{2}{n^2}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\prod_{k=1}^{m} \times\left[\frac{3}{2} - |x_{ik} - x_{jk}|(1 - |x_{ik} - x_{jk}|)\right], \tag{1}$$

where $x_i = (x_{i1}, \ldots, x_{im}) \in \mathcal{P}$. In this paper the set of points $\mathcal{P}$ is always chosen to be a $\tilde{U}(n; q^m)$ design, and we refer to WD_2-value of a $U(n; q^m)$ as the WD_2-value of its corresponding $\tilde{U}(n; q^m)$. Due to the specific structure of the U-type design, its corresponding uniformity measure WD_2 has some nice properties, which will be given in the next section.

A symmetrical uniform design under WD_2 is a U-type design $U(n; q^m)$, whose corresponding $\tilde{U}(n; q^m)$ minimizes WD_2-values over $\tilde{\mathcal{U}}(n; q^m)$. To search a uniform design is an NP hard problem in the sense of computational complexity when (n, q, m) increase. The threshold accepting algorithm has been used to search uniform designs by many authors such as [6,7,8]. The last paper showed that lower bounds of WD_2 play an important role in the search process. Unfortunately, their results are limited to $q = 2, 3$. For computer experiments uniform designs with $q > 3$ are often required. The task of the paper is to generalize their results from $U(n; q^m)$, $q = 2, 3$, to any symmetrical U-type designs. Meanwhile, combinatorial configurations have been extensively used to construct uniform designs under the discrete discrepancy (or categorical discrepancy) proposed by Hickernell and Liu [12]. Vital papers include [3–5,13,14]. In this paper, we will make use of the wrap-around L_2-discrepancy as the benchmark of uniformity to construct uniform designs via a new class of combinatorial configuration named "perfect resolvable balanced incomplete block design (PRBIBD)". Moreover, an efficient balance-pursuit heuristic for generating any symmetrical uniform design will also be proposed.

The paper is organized as follows: in Section 2, we give lower bounds of the wrap-around L_2-discrepancy on U-type uniform designs and also give corresponding sufficient conditions for a U-type design to achieve the lower bound. Based on these sufficient conditions, we provide two ways of construction methods for uniform designs in Section 3. First we deal with the combinatorial approach. The concept of PRBIBD, the connection between uniform designs and PRBIBDs and some results on the existence of infinite classes for PRBIBDs are explored in this section. We also introduce an efficient algorithm named balance-pursuit heuristic to search for uniform designs under the wrap-around L_2-discrepancy in Section 3. It is shown by several tests that our new algorithm is powerful. The last section addresses some conclusion and future work.

2. Sufficient conditions and lower bounds of WD_2

From the analytical expression of Eq. (1), it is easy to see that the wrap-around L_2-discrepancy is only a function of products of $\alpha_{ij}^k \equiv |x_{ik} - x_{jk}|(1 - |x_{ik} - x_{jk}|)$ $(i, j = 1, \ldots, n, i \neq j$ and $k = 1, \ldots, m)$. However, for a U-type design, its α-values can only be limited to a specific set. More precisely, for a U-type design $\tilde{U}(n; q^m)$, when q is even, α-values can only take $q/2 + 1$ possible values, i.e., $0, 2(2q-2)/(4q^2), 4(2q-4)/(4q^2), \ldots, q^2/(4q^2)$; when q is odd, these products can only take $(q+1)/2$ possible values, i.e., $0, 2(2q-2)/(4q^2), 4(2q-4)/(4q^2), \ldots, (q-1)(q+1)/(4q^2)$. Table 1 gives the distribution of α-values over the set $\{\alpha_{ij}^k : 1 \leqslant i < j \leqslant n, 1 \leqslant k \leqslant m\}$, for both even and odd q. Note that given (n, q, m), this distribution is the same for each design in $\tilde{\mathcal{U}}(n; q^m)$. We shall see that this fact is very useful in our approach.

For any two different rows of the design $\tilde{U}(n; q^m)$, $x_i = (x_{i1}, x_{i2}, \ldots, x_{im})$, and $x_j = (x_{j1}, x_{j2}, \ldots, x_{jm})$, denote by F_{ij}^{α} the distribution of their $\{\alpha_{ij}^k, k = 1, \ldots, m\}$. The F_{ij}^{α}'s can characterize whether a U-type design is a uniform design or not.

Theorem 2.1. *A lower bound of the wrap-around L_2-discrepancy on $\tilde{\mathcal{U}}(n; q^m)$ with even q and odd q is given by*

$$LB_{\text{even}} = \Delta + \frac{n-1}{n}\left(\frac{3}{2}\right)^{\frac{m(n-q)}{q(n-1)}}\left(\frac{5}{4}\right)^{\frac{mn}{q(n-1)}}\left(\frac{3}{2} - \frac{2(2q-2)}{4q^2}\right)^{\frac{2mn}{q(n-1)}} \cdots \left(\frac{3}{2} - \frac{(q-2)(q+2)}{4q^2}\right)^{\frac{2mn}{q(n-1)}};$$

$$LB_{\text{odd}} = \Delta + \frac{n-1}{n}\left(\frac{3}{2}\right)^{\frac{m(n-q)}{q(n-1)}}\left(\frac{3}{2} - \frac{2(2q-2)}{4q^2}\right)^{\frac{2mn}{q(n-1)}} \cdots \left(\frac{3}{2} - \frac{(q-1)(q+1)}{4q^2}\right)^{\frac{2mn}{q(n-1)}},$$

respectively, where $\Delta = -\left(\frac{4}{3}\right)^m + \frac{1}{n}\left(\frac{3}{2}\right)^m$. A U-type design $\tilde{U}(n; q^m)$ is a uniform design under the wrap-around L_2-discrepancy, if all its F_{ij}^{α} distributions, $i \neq j$, are the same. In this case, the WD_2-value of this design achieves the above lower bound.

Table 1
Distribution of α-values of a $U(n; q^m)$ design

q even		q odd	
α-values	Number	α-values	Number
0	$\frac{mn(n-q)}{2q}$	0	$\frac{mn(n-q)}{2q}$
$\frac{2(2q-2)}{4q^2}$	$\frac{mn^2}{q}$	$\frac{2(2q-2)}{4q^2}$	$\frac{mn^2}{q}$
$\frac{4(2q-4)}{4q^2}$	$\frac{mn^2}{q}$	$\frac{4(2q-4)}{4q^2}$	$\frac{mn^2}{q}$
...	...	...	...
...	...	...	...
$\frac{(q-2)(q+2)}{4q^2}$	$\frac{mn^2}{q}$	$\frac{(q-3)(q+3)}{4q^2}$	$\frac{mn^2}{q}$
$\frac{q^2}{4q^2}$	$\frac{mn^2}{2q}$	$\frac{(q-1)(q+1)}{4q^2}$	$\frac{mn^2}{q}$

Proof. By Eq. (1), to minimize $WD_2(\mathcal{P})^2$ over $\tilde{\mathcal{U}}(n; q^m)$ is equivalent to minimizing $\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \prod_{k=1}^{m} \left[\frac{3}{2} - \alpha_{ij}^k\right]$ with respect to α_{ij}^k's. From Table 1, we know that for given (n, q, m), the distribution of α-values is the same, so $\prod_{i=1}^{n-1} \prod_{j=i+1}^{n} \prod_{k=1}^{m} \left[\frac{3}{2} - \alpha_{ij}^k\right]$ is a constant on $\tilde{\mathcal{U}}(n; q^m)$ and $3/2 - \alpha_{ij}^k > 0$. Based on the geometric and arithmetic mean inequality, the WD_2-value arrives at its minimum if all $\prod_{k=1}^{m} \left[\frac{3}{2} - \alpha_{ij}^k\right]$ for $1 \leqslant i < j \leqslant n$ are the same. Obviously, the latter is the result when all F_{ij}^{α} are the same. The expression of the lower bound of the discrepancy is straightforward according to Table 1. $\square$

Applying Theorem 2.1 to a two-level $\tilde{U}(n; 2^m)$ design we find that its $\{\alpha_{ij}^k\}$ take only two possible values 0 and 1/4 with frequency $mn(n-2)/4$ and $mn^2/4$, respectively. In this case, each F_{ij}^{α} distribution can be uniquely determined by the Hamming distance, denoted by d_{ij}, between the ith row and the jth row of the design matrix. The Hamming distance between two rows is defined as the number of places where two rows take different values. The necessary condition that all F_{ij}^{α} distributions equal each other is equivalent to that each d_{ij} equals $m - \lambda_2$, where $\lambda_2 = m(n-2)/2(n-1)$. There are similar results for three-level U-type designs. For instance, for a three-level $\tilde{U}(n; 3^m)$ design its $\{\alpha_{ij}^k\}$ take two possible values 0 and 2/9 with frequency $mn(n-3)/6$ and $mn^2/3$, respectively. Let $\lambda_3 = m(n-3)/3(n-1)$. The necessary condition that all F_{ij}^{α} distributions are the same is equivalent to that all d_{ij}'s equal $m - \lambda_3$. In these two special cases, Theorem 2.1 can be simplified to the following corollary, which essentially can be found in [6].

Corollary 2.2. *A lower bound of the wrap-around L_2-discrepancy on $\tilde{\mathcal{U}}(n; q^m)$ with $q = 2$ and $q = 3$ is given by*

$$LB_2 = -\left(\frac{4}{3}\right)^m + \frac{1}{n}\left(\frac{3}{2}\right)^m + \frac{n-1}{n}\left(\frac{3}{2}\right)^{\lambda_2}\left(\frac{5}{4}\right)^{mn/2(n-1)}, \tag{2}$$

$$LB_3 = -\left(\frac{4}{3}\right)^m + \frac{1}{n}\left(\frac{3}{2}\right)^m + \frac{n-1}{n}\left(\frac{3}{2}\right)^{\lambda_3}\left(\frac{23}{18}\right)^{2mn/3(n-1)}, \tag{3}$$

respectively. A U-type design $U(n; 2^m)$ or $U(n; 3^m)$ is a uniform design under the wrap-around L_2-discrepancy if the Hamming distance of each pair of two distinct rows equals $m - \lambda_2$ when $q = 2$ or equals $m - \lambda_3$ when $q = 3$. In this case WD_2-value of the uniform design arrives at the above lower bound.

3. Constructions of uniform designs

As already mentioned in Section 1, combinatorial configurations have been proved to be very useful in the constructions of uniform designs, see for example [3,13,14] and references therein. In the process one always takes "discrete discrepancy" as the criterion of uniformity. However, discrete discrepancy is thought to be only a category discrepancy. That is, though it is defined under the high level case, it can only deal with one difference, due to its simple kernel function (see [12]). While for the wrap-around L_2-discrepancy, the design can distinguish $\lfloor \frac{q-1}{2} \rfloor$ differences among those q levels. Obviously, it can give more information compared with the discrete discrepancy for $q > 3$. The designs obtained by traditional combinatorial configurations such as RPBDs, RPDs and RCDs are not always uniform under the wrap-around L_2-discrepancy, though they are uniform under the discrete discrepancy. This suggests new combinatorial configurations are needed for constructions of uniform designs under WD_2. According to Theorem 2.1, when a U-type design achieves the lower bound LB_{even} or LB_{odd}, all its F_{ij}^{α} distributions, $i \neq j$, should be same. This leads us to define a new type of combinatorial configurations by adding extra constraints to a resolvable BIBD.

As usual, we call a *balanced incomplete block design* (BIBD) of index λ and order n a (n, k, λ)-BIBD. It is defined to be a pair $(V, \mathcal{B})$, where V is a set of n points and $\mathcal{B}$ is a family of subsets (called *blocks*) of V with size k, such that every pair of points of V occurs in exactly λ blocks. A BIBD is called *resolvable*, denoted by RBIBD, if its blocks can be partitioned into classes (called *parallel classes*), each being a partition of its point set. Let $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_m$ be parallel classes of a (n, k, λ)-RBIBD, $(V, \mathcal{B})$. Assign a certain order to the blocks in each $\mathcal{A}_i$ and then write $\mathcal{A}_i = \{B_1^i, B_2^i, \ldots, B_q^i\}$, where $q = n/k$. The unordered pair x, y is called *t-apart* in $\mathcal{A}_i$, if $x \in B_j^i$, $y \in B_{j+t}^i$ or $y \in B_j^i$, $x \in B_{j+t}^i$, where $j + t$ is taken modulo q and $t = 1, 2, \ldots, \lfloor q/2 \rfloor$. For convenience, when q is even and $t = q/2$, each $q/2$-apart is regarded as twice appearing in the same parallel class.

For a (n, k, λ)-RBIBD, it is readily calculated that $\lambda(n - 1) = m(k - 1)$, where m represents the number of parallel classes. Thus, the total number of t-apart, i.e., knm, is determined by n, k and λ.

Definition 3.1. A perfect resolvable balanced incomplete block design, denoted by $(n, k, \lambda; \mu)$-PRBIBD, is an RBIBD, $(V, \mathcal{B})$, such that for every $t = 1, 2, \ldots, \lfloor q/2 \rfloor$, each unordered pair $x, y \in V$ is t-apart in $\mu = \frac{2k}{k-1}\lambda$ parallel classes ($k \geqslant 2$).

Remark. When the block size k equals 1, then the definition of BIBD or RBIBD becomes trivial. However, in this case, we can still define the concept of t-apart. As an extension to Definition 3.1, if for every $t = 1, 2, \ldots, \lfloor q/2 \rfloor$, each unordered pair $x, y \in V$ is t-apart in exactly μ parallel classes, then we call a $(n, 1, 0)$-RBIBD a $(n, 1, 0; \mu)$-PRBIBD.

Example 3.2. A (12, 3, 2; 6)-PRBIBD can be formed by the following 11 parallel classes: $\mathcal{A}_0 = \{0, 1, 3\}, \{2, 6, 8\}, \{4, 5, 9\}, \{7, 10, \infty\}$ and

$$\begin{aligned}\mathcal{A}_i = \mathcal{A}_0 + i = &\{i, 1+i, 3+i\}, \{2+i, 6+i, 8+i\},\\ &\{4+i, 5+i, 9+i\}, \{7+i, 10+i, \infty\}.\end{aligned}$$

Here $i = 1, 2, \ldots, 10$ and the addition is taken modulo 11.

Let us now describe the link between PRBIBDs and uniform designs under the wrap-around L_2-discrepancy.

Given a $(n, k, \lambda; \mu)$-PRBIBD, $(V, \mathcal{B})$, where $V = \{1, 2, \ldots, n\}$. For each parallel class $\mathcal{A}_i = \{B_1^i, B_2^i, \ldots, B_q^i\}$ $(i = 1, 2, \ldots, m)$, construct a q-level column $d^i = (d_{li})$ as follows: set $d_{li} = u$, if point l is contained in the uth block of $\mathcal{A}_i$, B_u^i. Then the m columns constructed from $\mathcal{A}_i$ of $\mathcal{B}$ $(i = 1, 2, \ldots, m)$ form an experimental design with n runs and m factors. The level number of each factor is q.

Example 3.3. The following U-type design $U(12; 4^{11})$ is derived from the (12, 3, 2; 6)-PRBIBD in Example 3.2:

Row	1	2	3	4	5	6	7	8	9	0	1
0	1	4	3	2	4	2	3	3	1	2	1
1	1	1	4	3	2	4	2	3	3	1	2
2	2	1	1	4	3	2	4	2	3	3	1
3	1	2	1	1	4	3	2	4	2	3	3
4	3	1	2	1	1	4	3	2	4	2	3
5	3	3	1	2	1	1	4	3	2	4	2
6	2	3	3	1	2	1	1	4	3	2	4
7	4	2	3	3	1	2	1	1	4	3	2
8	2	4	2	3	3	1	2	1	1	4	3
9	3	2	4	2	3	3	1	2	1	1	4
10	4	3	2	4	2	3	3	1	2	1	1
∞	4	4	4	4	4	4	4	4	4	4	4

Further, we can prove that:

Theorem 3.4. *The experimental design $U(n; q^m)$ derived from a $(n, k, \lambda; \mu)$-PRBIBD is a uniform design under the wrap-around L_2-discrepancy.*

Proof. According to Theorem 2.1, it suffices to prove that for each pair of distinct rows i and j in the derived experimental design $U(n; q^m)$, the F_{ij}^{α} distributions are the same. The pair of distinct rows in the experimental design corresponds to the pair of distinct points in the point set of the PRBIBD. Moreover, if the two distinct points i and j appear in the same block of the kth parallel class, then in the derived experimental design $U(n; q^m)$, the kth α-value between rows i and j, α_{ij}^k, will be 0; if the pair of two distinct points i and j

is t-apart in the kth parallel class, $t = 1, 2, \ldots, \lfloor q/2 \rfloor$, then in the derived experimental design $U(n; q^m)$, the kth α-value between rows i and j, α_{ij}^k, will be $\frac{t}{q}\left(1 - \frac{t}{q}\right)$. From the definition of PRBIBD, we know that each pair of distinct points appears in exact λ blocks, and is t-apart in exact μ parallel classes. This ensures the derived experimental design $U(n; q^m)$ satisfies the condition in Theorem 2.1. □

What Theorem 3.4 says is that if we happen to have a PRBIBD, then we can obtain a uniform design under WD_2. Though, it is a very difficult task to construct a PRBIBD due to its complicated structure, we succeed in finding some classes, which we state in stages below and leave the proofs in the appendix.

Theorem 3.5. *For any prime* $n = q_1 q_2 - 1$, *there exists a* $(n+1, q_1, (q_1-1)(n-1); 2q_1(n-1))$-PRBIBD, *hence we can obtain a uniform design* $U(n+1; q_2^{n(n-1)})$ *under* WD_2.

Theorem 3.6. *For any prime* $n = q_1 q_2 - 1 \equiv 3 \pmod 4$, *there exists a* $(n+1, q_1, (q_1 - 1)(n-1)/2; q_1(n-1))$-PRBIBD, *hence we can obtain a uniform design* $U(n+1; q_2^{\frac{n(n-1)}{2}})$ *under* WD_2.

Theorem 3.7. *For any odd prime* q, *there exists a* $(q, 1, 0; 2)$-PRBIBD, *hence we can obtain a uniform design* $U(q; q^{q-1})$ *under* WD_2.

Our next result is established by using the cyclotomic classes of a finite field of prime order. Let $p = ef + 1$ be a prime, and ω be an arbitrary primitive root modulo p. Denote the unique multiplicative subgroup of order f in Z_p by C_0^e, and write C_j^e for $j = 0, 1, \ldots, e-1$ for the cosets of C_0^e in Z_p^*, namely, the cyclotomic classes of index e of Z_p. An e-subset S of Z_p is called a representative system for $Z_p \bmod C_0^e$, if S covers exactly one element in every cyclotomic class C_j^e $(0 \leqslant j \leqslant e-1)$.

Theorem 3.8. *For any prime* $n = 12m + 7$ *and* $m \not\equiv 1 \pmod 3$, *denote* $V = Z_n \cup \infty$ *and* ε *as a root of* $x^2 + x + 1 = 0$. *If there exist the following base blocks*:

$$\begin{aligned}
A_1 &= \{a_1, a_2, \ldots, a_{3m+2}\},\\
A_2 &= \{\varepsilon a_1, \varepsilon a_2, \ldots, \varepsilon a_{3m+2}\},\\
A_3 &= \{\varepsilon^2 a_1, \varepsilon^2 a_2, \ldots, \varepsilon^2 a_{3m+2}\},\\
A_4 &= \{a_{3m+3}, \varepsilon a_{3m+3}, \varepsilon^2 a_{3m+3}, a_{3m+4}, \varepsilon a_{3m+4}, \varepsilon^2 a_{3m+4},\\
&\quad \ldots, a_{4m+2}, \varepsilon a_{4m+2}, \varepsilon^2 a_{4m+2}, 0, \infty\}
\end{aligned}$$

such that

1. $\{a_1, a_2, \ldots, a_{4m+2}\}$ *is a representative system for* $Z_n \bmod C_0^{4m+2}$,
2. *differences occur in* $A_3 - A_1$ *and* $A_4 - A_2$ *form some representative systems of* $Z_n \bmod C_0^{4m+2}$, *where* $A_i - A_j = \{b_i - c_j : b_i \in A_i, c_j \in A_j\}$,

then there exists a $(n+1, 3m+2, (3m+1)(2m+1); 2(3m+2)(2m+1))$-PRBIBD, *hence we can obtain a uniform design* $U(n+1; 4^{(2m+1)n})$ *under* WD_2.

For all primes less than 100 which satisfy the conditions in Theorem 3.8, we find the desired base blocks A_1, A_2, A_3 and A_4 by a simple computer search. The result is listed in Table 2.

Example 3.9. The following uniform design $U(8; 4^7)$ is derived from the above base blocks for $n = 7$:

Row	1	2	3	4	5	6	7
0	4	2	3	3	1	2	1
1	1	4	2	3	3	1	2
2	2	1	4	2	3	3	1
3	1	2	1	4	2	3	3
4	3	1	2	1	4	2	3
5	3	3	1	2	1	4	2
6	2	3	3	1	2	1	4
∞	4	4	4	4	4	4	4

For given (n, q, m), the lower bounds in Theorem 2.1 cannot always be reached. As a matter of fact, it is often difficult to make a judgement whether the WD_2-value of the design is the smallest on $\tilde{\mathcal{U}}(n; q^m)$. In this case a resulted design by some powerful optimization algorithm is called a *low-discrepancy design* that might be a uniform design sometimes. The threshold accepting heuristic has been successfully applied for searching low-discrepancy designs by [6,7,8,17]. However, Theorem 2.1 provides not only the lower bounds that can be used for a benchmark, but also the importance of balance of $\{F_{ij}^{\alpha}\}$. Checking all F_{ij}^{α} distributions to be the same needs a heavy computational load. Therefore, we define

$$\delta_{ij} = \sum_{k=1}^{m} \ln\left(\frac{3}{2} - \alpha_{ij}^{k}\right),$$

for any two rows i and j. Obviously, for any $1 \leqslant i \neq j, p \neq q \leqslant n$ the fact that $F_{ij}^{\alpha} = F_{pq}^{\alpha}$ implies $\delta_{ij} = \delta_{pq}$, but the inverse may not be true. Aiming to adjust those δ_{ij}'s as equally as possible, we propose a more powerful algorithm, which is named *balance-pursuit heuristic*. Compared with the existing threshold accepting heuristic, for example [6], our algorithm has more chances to generate better designs in the sense of lower WD_2 in each iteration, since it gives an approximate direction to the better status, which can save considerable time in the computational searching. Moreover, our algorithm does not need the threshold accepting series, which actually plays an important role in the threshold accepting heuristic. As stated in [17], the aim of using a temporary worsening up to a given threshold value is to avoid getting stuck into a local minimum. But how to determine a proper threshold accepting series is itself a difficult problem, since it will depend on the structure and property of the design. In our algorithm, we also use a randomly warming-up procedure but different way to jump out from a local minimum. Details can be seen in the following discussion.

Similar to the threshold accepting heuristic, our algorithm is started with a randomly generated U-type design D^0. Then it will go into a large number, say τ, times of iteration. In each iteration the algorithm tries to replace the current solution D^{c} with a new one. The new design D^{new} is generated in a given neighborhood of the current solution D^{c}.

Table 2
Base blocks in Theorem 3.8

$n = 7$	$A_1 =$	{ 3 1 }
	$A_2 =$	{ 6 2 }
	$A_3 =$	{ 5 4 }
	$A_4 =$	{ 0 ∞ }
$n = 31$	$A_1 =$	{ 19 13 5 27 26 4 23 9 }
	$A_2 =$	{ 10 15 1 24 30 7 17 8 }
	$A_3 =$	{ 2 3 25 11 6 20 22 14 }
	$A_4 =$	{ 29 12 21 28 18 16 0 ∞ }
$n = 43$	$A_1 =$	{ 11 32 27 38 28 41 37 2 1 4 30 }
	$A_2 =$	{ 9 34 26 35 19 14 42 29 36 15 5 }
	$A_3 =$	{ 23 20 33 13 39 31 7 12 6 24 8 }
	$A_4 =$	{ 10 16 17 21 25 40 18 3 22 0 ∞ }
$n = 67$	$A_1 =$	{ 14 6 39 8 26 29 62 50 27 15 18 44 32 47 52 11 35 }
	$A_2 =$	{ 49 21 36 28 24 1 16 41 61 19 63 20 45 64 48 5 22 }
	$A_3 =$	{ 4 40 59 31 17 37 56 43 46 33 53 3 57 23 34 51 10 }
	$A_4 =$	{ 58 2 7 30 38 66 25 54 55 65 60 9 42 13 12 0 ∞ }
$n = 79$	$A_1 =$	{ 60 42 71 57 51 1 41 39 78 15 4 32 77 52 75 63 7 46 9 76 }
	$A_2 =$	{ 37 18 53 47 67 23 74 28 56 29 13 25 33 11 66 27 3 31 49 10 }
	$A_3 =$	{ 61 19 34 54 40 55 43 12 24 35 62 22 48 16 17 68 69 2 21 72 }
	$A_4 =$	{ 58 70 30 64 50 44 59 14 6 20 65 73 36 38 5 26 45 8 0 ∞ }

In fact, a neighborhood is a small perturbation of D^c. Difference of discrepancy between D^{new} and D^c is calculated and compared in each iteration. If the result is not worse, or the design needs to be warmed-up, then we replace D^c with D^{new} and continue the iteration. algorithm: During this process, the determination of the neighborhood is important. Most authors choose a neighborhood of the current solution D^c in a way such that each design in the neighborhood is still a U-type one. This requirement can be easily fulfilled by selecting one column of D^c and exchanging two elements in the selected column. To enhance convergence speed we use two possible ways of pre-selection methods in our program to determine the neighborhood choice, instead of using random selection elements within a column for exchanging as done in the literature. According to Theorem 2.1, we should reduce differences among the current δ_{ij}'s. So our two pre-selection methods both aim to distribute the distances δ_{ij}'s as evenly as possible. The first method is called "*maximal and minimal distances of row pairs*". Denote by $(\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2})$ and $(\boldsymbol{x}_{j_1}, \boldsymbol{x}_{j_2})$ the respective row pairs with maximal and minimal distances for the current design D^c. We randomly select a row $\boldsymbol{x}_i$ from $\boldsymbol{x}_{i_1}$ or $\boldsymbol{x}_{i_2}$ and a row $\boldsymbol{x}_j$ from $\boldsymbol{x}_{j_1}$ or $\boldsymbol{x}_{j_2}$. between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Then randomly select a column k; if the kth element, x_{ik}, in the row $\boldsymbol{x}_i$ is not equal to x_{jk}, the kth element in the row $\boldsymbol{x}_j$, then exchange x_{ik} and x_{jk} to obtain a new design D^{new}. If the difference ($\nabla = WD_2(D^{\text{new}}) - WD_2(D^{\text{c}})$) is non-positive, then replace the current design D^{c} with D^{new}; if ∇ is positive, which means the new design D^{new} is worse than the current design D^{c}, then we randomly produce an integer variable v with value $0 \sim 999$; if say $v < 3$, then let $D^{\text{c}} := D^{\text{new}}$. This procedure provides 0.3% probabilities to warm up when the new design becomes worse, which can help the program avoid getting stuck into a local minimum. Moreover, experience shows that a local minimum is always surrounded by many others, so our program will seldom drop into an endless loop. After being warmed up from a local minimum, it will reach another one with large probability, and thus can make our searching move ahead. Another pre-selected method is called "*single row with maximal and minimal sum of distances*". Based on row-pairwise distances of the current design D^{c}, we find a single row with maximal and another single row with minimal sum of distances, say row $\boldsymbol{x}_i$ and row $\boldsymbol{x}_j$. This means $\sum_{t \neq i} \delta_{ti}$ is maximal and $\sum_{t \neq j} \delta_{tj}$ is minimal among $\sum_{t \neq k} \delta_{tk}, k = 1, \ldots, m$. Now randomly select a column k; if x_{ik}, the kth element in the row $\boldsymbol{x}_i$, is not equal to x_{jk}, the kth element in the row $\boldsymbol{x}_j$, then exchange x_{ik} and x_{jk} to obtain a new design D^{new}. Calculate the difference $\nabla = WD_2(D^{\text{new}}) - WD_2(D^{\text{c}})$, and perform the same record and replace procedure as stated in the first pre-selection method. For each iteration in our program, we randomly select a method and use them alternatively. Each method has its own advantages. Compared with "maximal and minimal distances of row pairs" method, "single row with maximal and minimal sum of distances" method is expected to accelerate the searching more, while the former method can provide more chances of jumping out from a local minimal status. The main idea of randomly using these two pre-selection methods to determine the neighborhood for each iteration is both to accelerate the speed and to jump out from a local minimal status. And experiments also show that when a single pre-selection method is used, the result will always be worse.

For accelerating the speed, our program also incorporates with other techniques. Instead of calculating two discrepancies of $WD_2(D^{\text{new}})$ and $WD_2(D^{\text{c}})$, Fang et al. [6] focused on the difference between $WD_2(D^{\text{new}})$ and $WD_2(D^{\text{c}})$. Our program also takes this advantage. Based on Eq. (1), we know that the wrap-around L_2-discrepancy can be expressed in terms of the sum of $e^{\delta_{ij}}$'s. And for a single exchange of two elements in the selected column, there are altogether $2(n-2)$ distances (δ_{ij}'s) updated. Suppose the kth elements in rows $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are exchanged; then for any row $\boldsymbol{x}_t$ other than $\boldsymbol{x}_i$ or $\boldsymbol{x}_j$, the distances of row pair $(\boldsymbol{x}_i, \boldsymbol{x}_t)$ and row pair $(\boldsymbol{x}_j, \boldsymbol{x}_t)$ will be changed. Denote $\tilde{\delta}_{ti}$ and $\tilde{\delta}_{tj}$ as the new distances between row pair $(\boldsymbol{x}_i, \boldsymbol{x}_t)$ and row pair $(\boldsymbol{x}_j, \boldsymbol{x}_t)$; then

$$\tilde{\delta}_{it} = \delta_{it} + \ln(3/2 - \alpha_{jt}^k) - \ln(3/2 - \alpha_{it}^k),$$

$$\tilde{\delta}_{jt} = \delta_{jt} + \ln(3/2 - \alpha_{it}^k) - \ln(3/2 - \alpha_{jt}^k).$$

Here α_{it}^k and α_{jt}^k are α-values as defined in Section 2. And the objective function change will be

$$\nabla = \sum_{t \neq i,j} \left(e^{\tilde{\delta}_{it}} - e^{\delta_{it}} + e^{\tilde{\delta}_{jt}} - e^{\delta_{jt}} \right).$$

Moreover, during the iteration, as soon as the lower bound is reached, the process will be terminated. But the lower bound may not be reached in many cases. For example, for achieving the lower bound in Theorem 2.1 when q is even, the numbers $\frac{mn(n-q)}{2q}$ and $\frac{mn^2}{2q}$ should be a multiple of $\frac{n(n-1)}{2}$ (cf. Table 1). Let $n = tq$ because n is a multiple of q. The necessary condition is equivalent to that $(n-1)$ is a devisor of both $m(t-1)$ and mt that is equivalent to that $(n-1)$ is a devisor of m. Therefore, Theorem 2.1 gives lower bounds only for $\tilde{\mathcal{U}}(n; q^m)$ where $(n-1)$ is a devisor of m when q is even. Similar constraints can also be deduced from Table 1 for q odd. When these basic constraints are not satisfied, the lower bound in Theorem 2.1 can never be achieved, so the termination of the iteration process will be controlled by the number of iterations. The following gives a pseudo-code of our proposed balance-pursuit heuristic. When the above basic constraints are not satisfied, then step 5 is not necessary.

Algorithm 1 Searching uniform designs under WD_2

1: Initialize τ
2: Generate starting design $D^{\mathrm{c}} \in \mathcal{U}(n, q^m)$ and let $D^{\min} := D^{\mathrm{c}}$
3: **for** $i = 1$ to τ **do**
4:   Generate $D^{\mathrm{new}} \in \mathcal{N}(D^{\mathrm{c}})$ by randomly using two pre-selection methods
5:   **if** $WD_2(D^{\mathrm{new}})$ achieve the lower bound **then** return(D^{new}) **end if**
6:   **if** $WD_2(D^{\mathrm{new}}) \leqslant WD_2(D^{\mathrm{c}})$ **then**
7:     $D^{\mathrm{c}} := D^{\mathrm{new}}$
8:     **if** $WD_2(D^{\mathrm{new}}) < WD_2(D^{\min})$ **then** $D^{\min} := D^{\mathrm{new}}$ **end if**
9:   **else if** rand(1000)< 3 **then** $D^{\mathrm{c}} := D^{\mathrm{new}}$
10:   **end if**
11: **end for**
12: return($D^{\min}$)

For testing performance of our new algorithm we consider the following two ways:

(A) *Comparison with existing results*: Fang et al. [6] obtained a number of new low-discrepancy designs with three levels by using at most 50,000,000 iterations. By the new algorithm with at most 25,000,000 iterations (half of FLW's iterations), which averagely takes about 25 min under the Visual Studio environment on a personal computer for each design, we find many new designs with even lower discrepancy than theirs. Table 3 lists these designs. Though the improvement seems marginal, some of these results have theoretical importance. For example, the new design $U_{18}(3^6)$ in Table 3 is an orthogonal array with strength two. Notice that the design $U(18; 3^6)$ listed in [6] is also an orthogonal array with strength two; we again get the assertion that orthogonal designs may not always be uniform. However, since our design $U(18; 3^6)$ has achieved the lower bound 0.4961, it is actually a *uniform orthogonal design*. So far there are very few uniform orthogonal designs in the literature.

Table 3
Low-discrepancy designs with three levels and comparison with results in [6]

Row	Column	Res. (FLW)	Results	Row	Column	Res. (FLW)	Results
12	11	5.5591	5.5506	21	12	6.9172	6.9011
15	8	1.3276	1.3254	21	15	23.9076	23.9044
15	9	2.0905	2.0876	21	17	53.9094	53.8928
15	10	3.2688	3.2681	24	9	1.9102	1.9082
15	11	5.0699	5.0644	24	10	2.9970	2.9096
15	12	7.8292	7.8234	24	12	6.6702	6.6566
15	13	12.0110	12.0020	24	13	10.0880	10.0423
15	17	64.3040	64.2707	24	14	15.1755	15.1586
15	21	333.6779	333.5764	24	15	22.7304	22.7284
18	6	0.4972	0.4961	27	9	1.8727	1.8688
18	8	1.2579	1.2517	27	10	2.8544	2.8495
18	10	3.0837	3.0633	27	12	6.4991	6.4873
18	11	4.7569	4.7531	27	13	9.7263	9.6912
21	8	1.2573	1.2569	27	14	14.5836	14.5792
21	9	1.9427	1.9413				

Note: The third column indicates the results listed in Fang [6], while the fourth column indicates the results obtained by our code.

Table 4
Deviation (in percent) from lower bound for designs with level 10

Row	Column	Results	Lower bound	Dev.
100	30	1683.551441	1599.720435	5.2404
100	31	2553.635741	2441.198723	4.6058
100	32	3869.932487	3719.204383	4.0527
100	33	5855.258673	5657.898236	3.4882
100	34	8849.818852	8595.697336	2.9564
100	35	13382.105508	13043.197930	2.5983
100	36	20207.595988	19770.347912	2.2116
100	37	30510.933107	29937.597121	1.9151
100	38	46027.871085	45293.155126	1.6221
100	39	69407.053583	68469.601192	1.3692
100	40	104621.889690	103429.781635	1.1526

Note: The third column indicates the results obtained by our code, the fourth column represents the lower bound in Theorem 2.1, while the last column shows the deviation percent.

(B) *Designs with a large number of runs*: Computer experiments need uniform designs with a large number of runs. All the designs in the web site "http://www.math.hkbu.edu.hk" are with a number of runs $\leqslant 42$ due to computation complexity. Table 4 lists WD_2-values for the designs with 10 levels, 100 runs and the number of factors from 30 to 40, obtained by the new method. The lower bounds in Theorem 2.1 can serve as benchmarks for measuring the designs obtained. Let deviation (Dev.) be the percent of $(WD_2(D^*) - \text{lower bound})/\text{lower bound}$, where D^* is the design obtained by our computational search. We can see that most deviations are less than 5%. Considering the lower bounds may not be reached in many cases, our experiment listed in Table 4 is very satisfactory.

4. Conclusion and future work

In our paper, we present explicit lower bounds for measuring uniformity of symmetrical U-type designs with any level under the wrap-around L_2-discrepancy. We also provide two methods, combinatorial and optimization approaches, for constructing uniform designs or low-discrepancy designs under the wrap-around L_2-discrepancy. Moreover, the basic techniques included in these methods may also be paralleled to deal with uniform designs under discrepancies other than WD_2. For example, we can utilize more existing or new defined combinatorial configurations to construct uniform designs under several different discrepancies, say the centered L_2-discrepancy, the symmetrical centered L_2-discrepancy or other discrepancies. In such a way, we can not only establish a connection between combinatorial designs and uniform designs, but also provide a platform to unify different discrepancies by using different properties of combinatorial configurations. On the other hand, modification and improvement to our balance-pursuit heuristic algorithm is also interesting. How to make the algorithm more efficient and more flexible to different discrepancies will always be helpful. So further investigation into these researches will be carried on.

Acknowledgments

We thank both referees for their suggestions, comments and remarks which greatly help us to improve the presentation of this paper. Fang's research was partially supported by the Hong Kong RGC Grants RGC/HKBU 2044/02P and RGC/HKBU 200804, and by Hong Kong Baptist University FRG Grant FRG/03-04/II-711, and Yin's research was supported by an NSF Grant 10371086.

Appendix

Proof of Theorem 3.5. Let $V = Z_n \cup \{\infty\}$ be the point set. Let ω be the primitive element of $GF^*(n)$. Here Z_n means the additive group modulo n, $GF^*(n)$ means the multiplicative group of finite field $GF(n)$ and the infinity ∞ indicates a fixed point, which added or multiplied by any element in V results in ∞ itself. Denote $\mathcal{A}_0 = \{B_1, B_2, \dots, B_{q_2}\}$ as any partition of V with each block size q_1. Define

$$\mathcal{A}_{il} = \omega^i \mathcal{A}_0 + l = \{\omega^i B_1 + l, \omega^i B_2 + l, \dots, \omega^i B_{q_2} + l\},$$
$$i = 0, 1, 2, \dots, n-2; \quad l = 0, 1, \dots, n-1.$$

Here $\omega^i B_j + l = \{\omega^i b + l : b \in B_j\}$ and $\omega^i \infty + l = \infty$. It can be easily checked that $\mathcal{A}_{il}$s ($i = 0, 1, \dots, n-2$, $l = 0, 1, \dots, n-1$) form $n(n-1)$ parallel classes of a PRBIBD. The parameters for both PRBIBD and derived uniform design can then be calculated straightforward. $\square$

Proof of Theorem 3.6. We use the same notations as in Theorem 3.5, but define

$$\mathcal{A}_{il} = \{\omega^{2i} B_1 + l, \omega^{2i} B_2 + l, \ldots, \omega^{2i} B_{q_2} + l\},$$
$$i = 0, 1, 2, \ldots, \frac{n-3}{2}; \quad l = 0, 1, \ldots, n-1.$$

Here $\omega^{2i} B_j + l = \{\omega^{2i} b + l : b \in B_j\}$ and $\omega^{2i}\infty + l = \infty$. Then notice -1 is a quadratic nonresidue $(\mathrm{mod}\, n)$ for $n \equiv 3\,(\mathrm{mod}\, 4)$; we know that $\mathcal{A}_{il}$'s ($i = 0, 1, \ldots, \frac{n-3}{2}$, $l = 0, 1, \ldots, n-1$) form $\frac{n(n-1)}{2}$ parallel classes of a PRBIBD. The parameters for both PRBIBD and derived uniform design can then be calculated straightforward. □

Proof of Theorem 3.7. Let Z_p, the additive group modulo n, be the point set V. Define

$$\mathcal{A}_i = \{a_{i0}, a_{i1}, \ldots, a_{i(q-1)}\}, \quad i = 0, 1, \ldots, q-2.$$

Here $a_{ij} = (i+1) \times j\,(\mathrm{mod}\, q)$ ($j = 0, 1, \ldots, q-1$). It can be checked that $\mathcal{A}_i$s ($i = 0, 1, \ldots, q-2$) form $q-1$ parallel classes of a PRBIBD. The parameters for both PRBIBD and derived uniform design can then be calculated straightforward. □

Proof of Theorem 3.8. Since ε is a root of $x^2 + x + 1 = 0$, the multiplicative subgroup spanned by ε is C_0^{4m+2}. While $\{a_1, a_2, \ldots, a_{4m+2}\}$ is a representative system of $Z_n \bmod C_0^{4m+2}$, so A_1, A_2, A_3 and A_4 form a parallel class, which partitions V. Denote $\mathcal{A}_0 = \{A_1, A_2, A_3, A_4\}$ and define

$$\mathcal{A}_{cl} = \{cA_1 + l, cA_2 + l, cA_3 + l, cA_4 + l\}, \quad c \in C_0^6, \;\; l \in Z_n,$$

where $cA_i + l = \{cb_j + l : b_j \in A_i\}$ and $c\infty + l = \infty$. Since $m \not\equiv 1\,(\mathrm{mod}\, 3)$, we can know that ± 1, $\pm\varepsilon$ and $\pm\varepsilon^2$ form a representative system of $Z_n \bmod C_0^6$. So it is easy to check that every pair of distinct points occurs together in exactly $(3m+1)(2m+1)$ blocks in those $\mathcal{A}_{cl}$'s. Based on the second condition and also noticing -1 is a quadratic nonresidue, we know that every pair of distinct points is 2-apart in exactly $2(3m+2)(2m+1)$ parallel classes, thus is also 1-apart in exactly $n(2m+1) - (3m+1)(2m+1) - (3m+2)(2m+1) = 2(3m+2)(2m+1)$ parallel classes. So those $\mathcal{A}_{cl}$'s form the desired PRBIBD. The parameters for the derived uniform design can then be calculated straightforward. □

References

[1] R.A. Bates, R.J. Buck, E. Riccomagno, H.P. Wynn, Experimental design and observation for large systems, J. Roy. Statist. Soc. Ser. B 58 (1996) 77–94.

[2] K.T. Fang, The uniform design: application of number-theoretic methods in experimental design, Acta Math. Appl. Sinica 3 (1980) 363–372.

[3] K.T. Fang, G.N. Ge, M.Q. Liu, Uniform supersaturated design and its construction, Sci. in China (A) 45 (2002) 1080–1088.

[4] K.T. Fang, G.N. Ge, M.Q. Liu, H. Qin, Construction on minimum generalized aberration designs, Metrika 57 (2003) 37–50.

[5] K.T. Fang, X. Lu, Y. Tang, J.X. Yin, Constructions of uniform designs by using resolvable packings and coverings, Discrete Math. 274 (2004) 25–40.

[6] K.T. Fang, X. Lu, P. Winker, Lower bounds for centered and wrap-around L_2-discrepancies and construction of uniform design by threshold accepting, J. Complexity 19 (2003) 692–711.

[7] K.T. Fang, C.X. Ma, Wrap-around L_2-discrepancy of random sampling, Latin hypercube and uniform designs, J. Complexity 17 (2001) 608–624.
[8] K.T. Fang, C.X. Ma, P. Winker, Centered L_2-discrepancy of random sampling and Latin hypercube design, and construction of uniform designs, Math. Comp. 71 (2002) 275–296.
[9] K.T. Fang, Y. Wang, Number-Theoretic Methods in Statistics, Chapman & Hall, London, 1994.
[10] F.J. Hickernell, A generalized discrepancy and quadrature error bound, Math. Comp. 67 (1998) 299–322.
[11] F.J. Hickernell, Lattice rules: how well do they measure up?, in: P. Hellekalek, G. Larcher (Eds.), Random and Quasi-Random Point Sets, Springer, Berlin, 1998, pp. 106–166.
[12] F.J. Hickernell, M.Q. Liu, Uniform designs limit aliasing, Biometrika 89 (2002) 893–904.
[13] M.Q. Liu, R.C. Zhang, Construction of $E(s^2)$ optimal supersaturated designs using cyclic BIBDs, J. Statist. Plann. Inference 91 (2000) 139–150.
[14] X. Lu, W.B. Hu, Y. Zheng, A systematical procedure in the construction of multi-level supersaturated design, J. Statist. Plann. Inference 115 (2003) 287–310.
[15] H. Niederreiter, Random number generation and quasi-Monte Carlo methods, SIAM CBMS-NSF Regional Conference Series in Applied Mathematics, Philadelphia, 1992.
[16] Y. Wang, K.T. Fang, A note on uniform distribution and experimental design, KeXue TongBao 26 (1981) 485–489.
[17] P. Winker, K.T. Fang, Optimal U-type design, in: H. Niederreiter, P. Zinterhof, P. Hellekalek (Eds.), Monte Carlo and Quasi-Monte Carlo Methods 1996, Springer, Berlin, 1998, pp. 436–448.